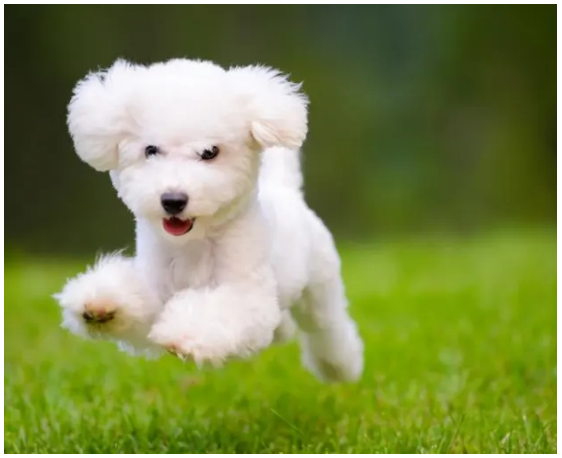
Image Segmentation by Normalised Cut

Computational Steps

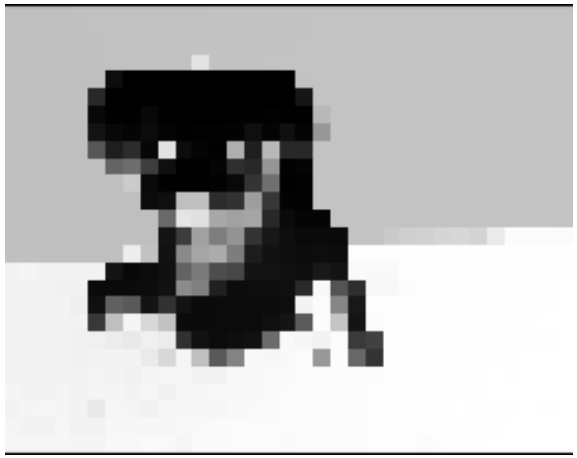
- **Preprocessing**: Each image is resized to 64X64
- **Feature Selection** :
  Feature 1>Euclidean Distance
  Feature 2>Grey Intensity
- Calculation of **Similarity** and **Degree Matrix** (4096X4096)
- Calculation of **Eigenvectors** corresponding to **2nd smallest** eigenvalue
- **Thresholding**: Quantising the values into discrete levels according to number of segments
- Display: Reshape the 4096X1 eigenvector to 64X64 image and display the image

Image 1 results



Original Image

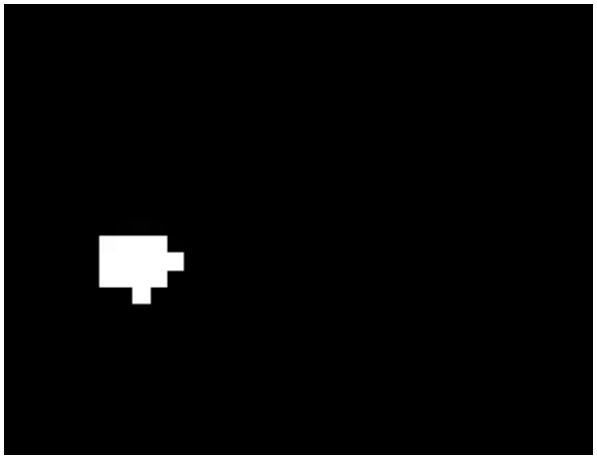
Feature 1



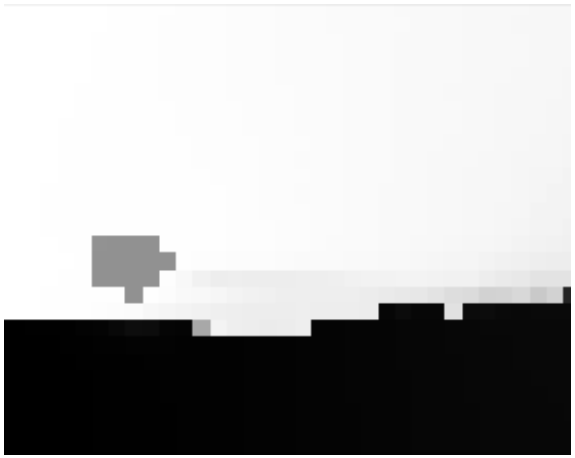
Feature 2

Image 2 results

Original Image



Feature 1



Feature 2

Image 3 results

Original Image

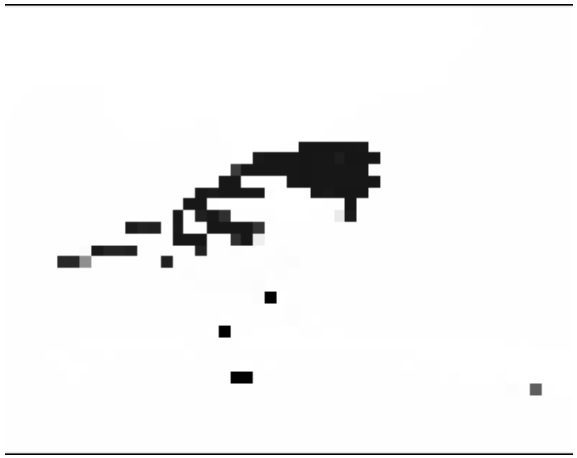
Feature 1

Feature 2

Image 4 results

Original Image

Feature 1



Feature 2

Takeaways

- N-Cut algorithm is computationally expensive compared to K-Means clustering.
- N-cut algorithm is very sensitive to the value of threshold taken.

Image Segmentation by K-means

Implementation details

Features : Pixel Intensity

Cluster mean Initialisation: random

Max Iterations: 100

Image 1 results

Original Image

K = 3

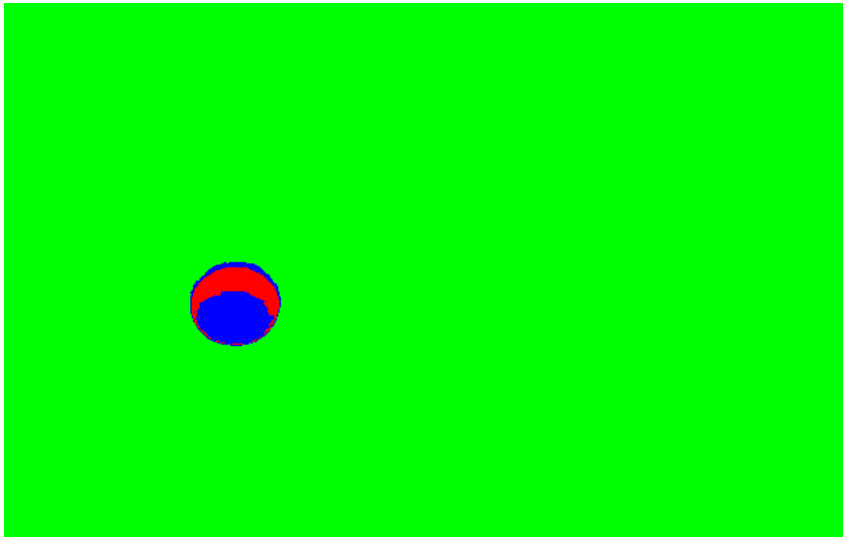
K = 4

K = 5

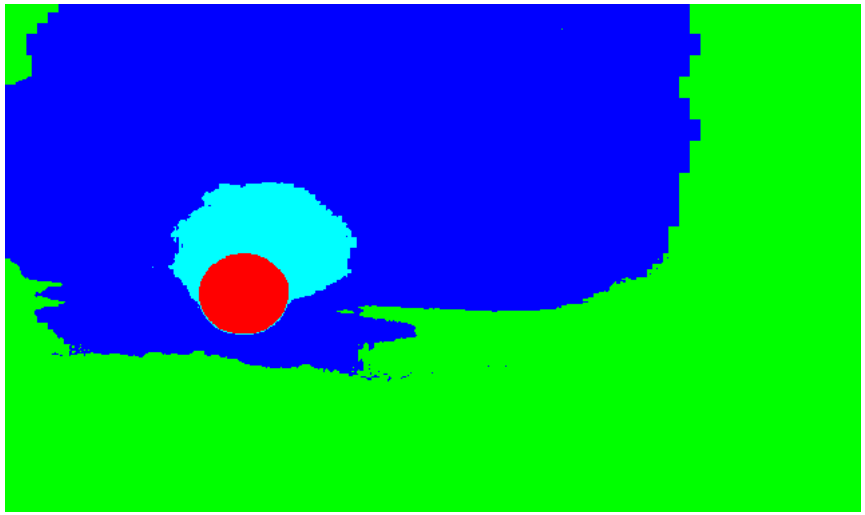
K = 6

Image 2 results

Original Image

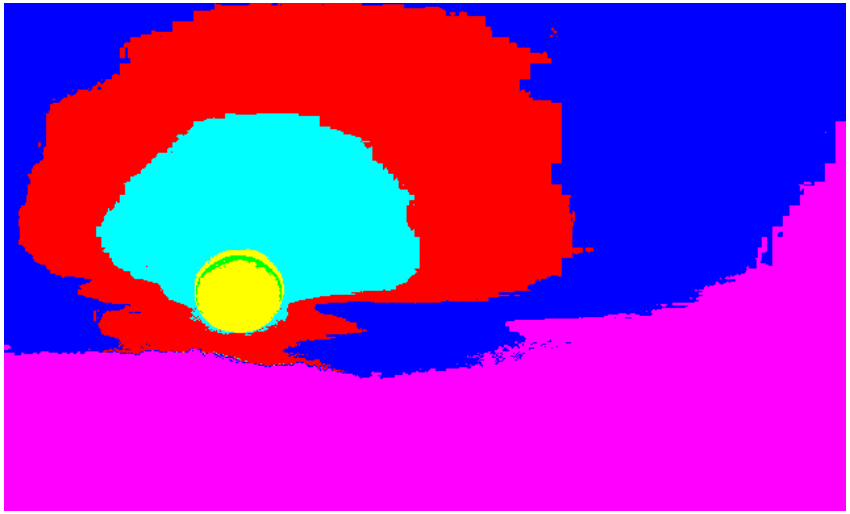


K = 3



K = 4

K = 5



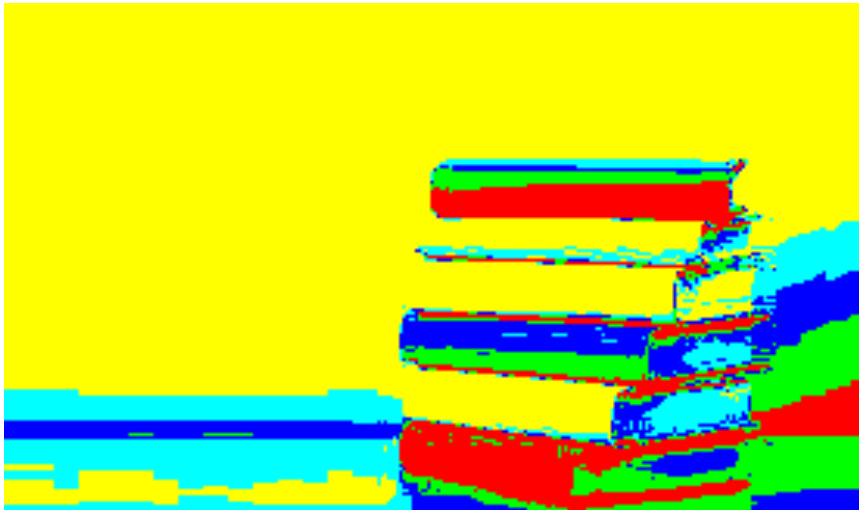
K = 6

Image 3 results

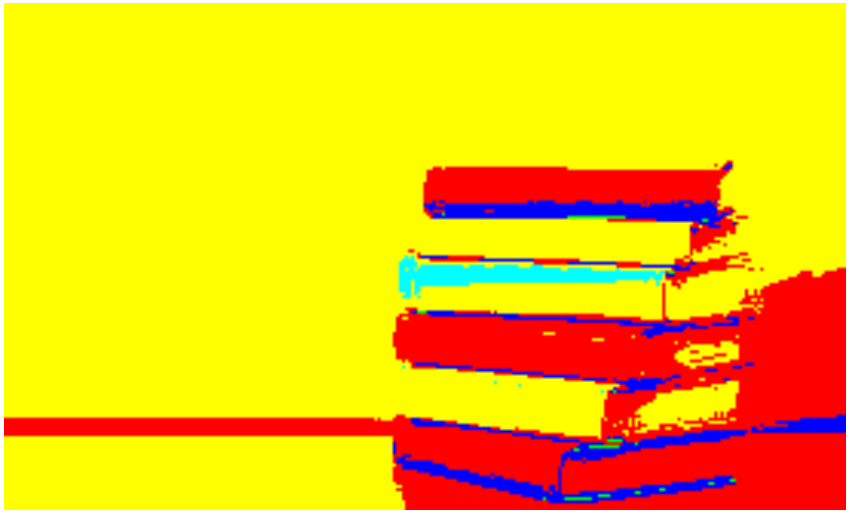
Original Image

K = 3

K = 4



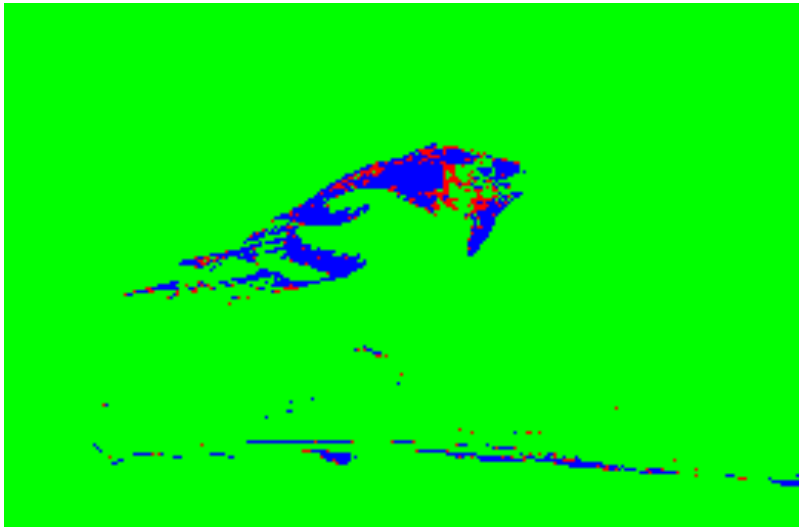
K = 5



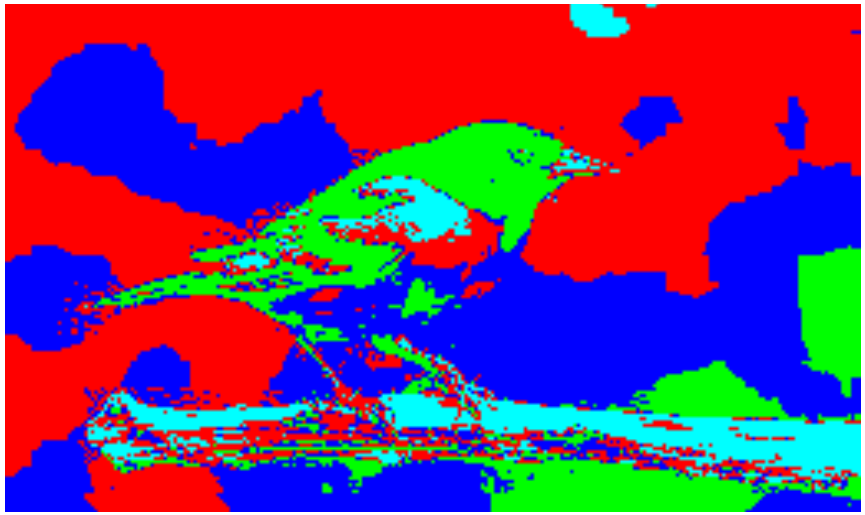
K = 6

Image 4 results

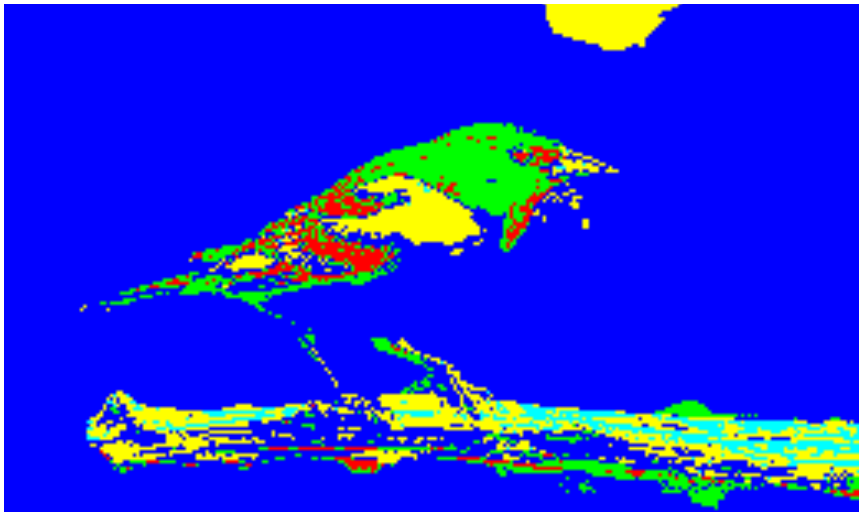
Original Image



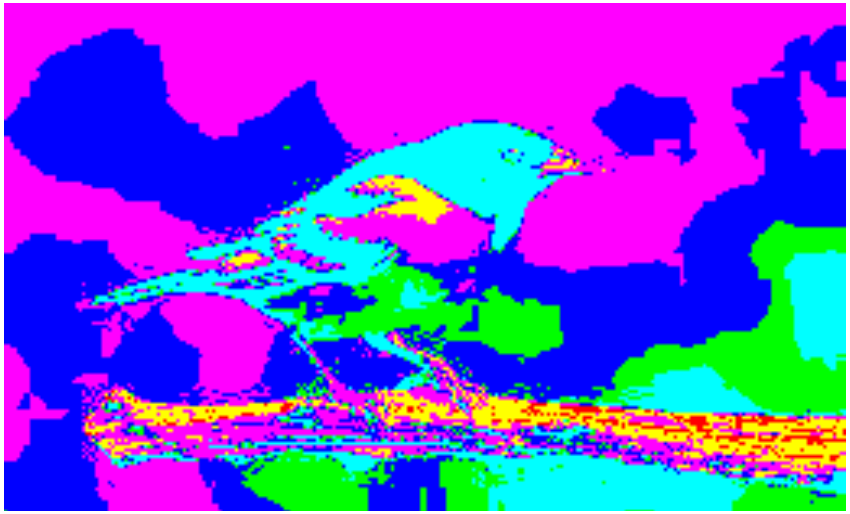
K = 3



K = 4



K = 5



K = 6

Takeaways

✳K-means algorithm is very sensitive to the choice of initial centroid. And also to the extending features size (I have tried making feature vector of 05 dim[R, G, B, x, y]), this heavily affected the image segmentation.

✳In case of K-means if numbers of clusters taken in algorithm are more then the numbers of different segments in the input image, the algorithm gives an over-segmented image.